On Two Factors Affecting the Efficiency of MILP Models in Automated Cryptanalyses

Shengyuan Xu[1,2], Xiutao Feng[1]✉, and Yongxing Wang[1,2]

[1] Key Laboratory of Mathematics Mechanization, Academy of Mathematics and Systems Science, Chinese Academy of Sciences;
[2] University of Chinese Academy of Sciences
fengxt@amss.ac.cn

Abstract. In recent years, mixed integer linear programming (MILP, in short) gradually becomes a popular tool of automated cryptanalyses in symmetric ciphers, which can be used to search differential characteristics and linear approximations with high probability/correlation. A key problem in the MILP method is how to build a proper model that can be solved efficiently in the MILP solvers like Gurobi or Cplex. It is known that a MILP problem is NP-hard, and the numbers of variables and inequalities are two important measures of its scale and time complexity. Whilst the solution space and the variables in many MILP models built for symmetric cryptanalyses are fixed without introducing dummy variables, the cardinality, i.e., the number of inequalities, is a main factor that might affect the runtime of MILP models. We notice that the norm of a MILP model, i.e., the maximal absolute value of all coefficients in its inequalities, is also an important factor affecting its runtime. In this work we will illustrate the effects of two parameters cardinality and norm of inequalities on the runtime of Gurobi by a large number of cryptanalysis experiments. Here we choose the popular MILP solver Gurobi and view it a black box, construct a large number of MILP models with different cardinalities or norms by means of differential analyses and impossible differential analyses for some classic block ciphers with SPN structure, and observe their runtimes in Gurobi. As a result, our experiments show that although minimizing the number of inequalities and the norm of coefficients might not always minimize the runtime, it is still a better choice in most situations.

Keywords: Automated cryptanalysis · Mixed integer linear programming · Full linear integer inequality characterization · Cardinality · Norm

1 Introduction

Mixed integer linear programming (MILP, in short) is a crucial optimization problem in operational research. It is a class of NP-hard problems whose goal is to minimize/maximize a linear objective function under linear constraints. As an effective mathematical modeling method to solve complex optimization tasks, MILP is widely used to solve various problems such as the assignment problem [1], production and distribution planning [2], optimal power flow problem [3], and cryptanalysis [4].

Differential analysis [5] and linear analysis [6] are two of the most important cryptanalyses in block ciphers, the core of them is to search for differential/linear trails with high probabilities/correlations to construct distinguishers or accomplish key recovery attacks. There are many effective variants of differential analysis and linear analysis such as truncated differential attack [7], related-key differential attack [8], impossible differential attack [9] and zero correlation attack [10]. Over the past decade years, MILP has become one of the most widely used tools in automated search algorithms that can effectively search for differential/linear trails with high probability/correlation.

The MILP-based method was first introduced into differential and linear analysis by Mouha *et al*. [4] to search the minimal number of active S-boxes. Later Sun *et al.* [11] extended the previous method to S-bP structures. In another work, Sun *et al.* [12] proposed an automatic method for evaluating the security of bit-oriented block ciphers against the (related-key) differential attack. After that, the MILP-based method has become more widely used. In [13], Fu *et al*. proposed a MILP-based method for searching differential characteristics and linear approximations for ARX ciphers. Combining the correlation of a quadratic Boolean function based on its disjoint quadratic form, Shi *et al.* [14] derived a MILP problem from a generic model for MORUS-like keystream generators and determined the correlations of linear trails of MiniMORUS and MORUS. MILP-based methods could also be used to search zero-correlation distinguishers [15] and impossible differential distinguishers [16].

Besides, the MILP-based method also has many applications in other cryptanalyses. In 2009, the MILP method was applied to solve a sparse system of quadratic equations and presented a numerical attack on some reduced versions of Trivium [17]. Albrecht and Cid [18] proposed a new algorithm to solve a set of nonlinear algebraic equations derived from cold boot attacks by constructing MILP models in 2011. Later, Walter *et al.* [19] optimized guessing strategies for algebraic cryptanalysis on the block cipher EPCBC with MILP models in 2013. Xiang *et al*. [20] translated the propagation of division property into a MILP problem and searched integral distinguishers. By modeling the division trails with MILP, the superpoly could be recovered in cube attacks [21]. In 2020, Cen *et al.* [22] proposed a new method to solve the minimizing problem of deduction systems based on MILP to search the optimal trail of guess and determine analysis.

In summary, MILP has gradually become a powerful tool for automated cryptanalyses in symmetric ciphers, and further exploration of the MILP problem has recently become a hot topic in cryptography. In MILP-based automated cryptanalyses, the common and crucial problem is the efficiency of solving target models. There are many mature solvers for MILP problems, such as Gurobi [23], Cplex [24], and Minisat [25]. Among them, the Gurobi optimizer is one of the best-performing commercial solvers for MILP problems, which is also used for experiments in this paper. However, the internal algorithms of Gurobi are not opened in detail, and users can not figure out which model is more efficient before calling Gurobi to solve it. Therefore how to construct an effective model for Gurobi attracted more and more attentions from researchers in recent years.

In general, a MILP model has two main parameters that determine the scale of the problem: the number of variables and the number of inequalities. As an NP-Hard problem, the dominant view is that the smaller the scale, the easier it is to solve. As for MILP models built for automated cryptanalyses in most cases, the values of variables are limited in $\mathbb{Z}_2$, and the coefficients of inequalities are restricted to $\mathbb{Z}$. Furthermore, for a same cryptanalytic object, the solution space and the number of variables of its corresponding MILP models are usually the same when no dummy variables are introduced. Consequently, the main difference for models of the same cryptanalysis is the number of inequalities, which will be called the cardinality of the inequality set later.

In the previous work, researchers always tended to reduce the number of inequalities to improve the efficiency of the model. However, this assertion was lack of experimental and theoretical support. In [26], Todo and Sasaki claimed that the folklore "minimizing the number of inequalities minimizes the runtime" is not always correct. They ran experiments with various numbers of inequalities to check how the number of inequalities is related to the runtime of MILP models. In addition, many other parameters also have effects on the efficiency of the solution: Li *et al.* [27] presented the influences between the construction and solution of MILP models solved by Gurobi, and considered the number of variables and constraints and the order of constraints and variables. Besides, according to some experimental results, we found that the maximal absolute value of all coefficients in inequalities, which will be called the norm later, also has a significant influence on the efficiency of the model.

In this paper, our main goal is to illustrate the effect of the norm and cardinality of MILP models on the runtime of Gurobi by conducting a large number of cryptanalysis experiments.

1.1 Related Work

There were lots of works to construct efficient models by reducing the number of inequalities to characterize a given set $S \subseteq \mathbb{Z}_2^n$. In 2014, Sun *et al.* [28] computed the H-representation of the convex hull of S with the mathematical software SAGE [29], then a greedy algorithm was applied to remove the redundancy and select as few inequalities as possible. Abdelkhalek *et al.* [30] constructed inequalities by logical conditions and converted the problem into a problem of minimizing the product-of-sum representation of Boolean functions to get fewer inequalities. Based on the Set Covering Problem (SCP, in short), Todo and Sasaki [16] proposed a MILP model to select the minimal number of inequalities from a given candidate set. In 2019, Li *et al.* proposed a new way to construct inequalities from a lower dimensional case. Using the inequalities obtained through the previous methods, they obtained fewer inequalities than before for larger dimensional cases. Based on the algebraic and geometric structure of sets, Boura *et al.*

[31] further reduced the number of inequalities to characterize a given set. By adding up inequalities in the candidate set and selecting inequalities using the SCP-based method, they characterized sets with fewer inequalities for $n \leq 10$. For larger n, a new structure of points in S that could be cut by the same inequality was explored, and better results were obtained. Later, properties of S that can be characterized by only one linear inequality were further studied. Udovenk [32] and Sun [33] respectively proposed new algorithms and updated the results of the previous work. Independently with Udovenk [32] and Sun [33], by exploring properties of plain sets such as type, sparsity, degeneration, order, minimal and maximal element, Feng *et al.* [34] established a complete theoretical system to solve the problem of fully characterizing a given set with the minimal number of inequalities. They provided a algorithm of enumerating all plain closures for a give S-box, which supports point sets with high dimension up to 18 and is the fastest at present. They first obtained all plain closures of many common S-Boxes used in block ciphers. As for the MILP characterizations of many common S-Boxes, all results they got are the best at present. Particularly, their results are far better than the previous in the high dimensional cases.

With regard to the norm of inequality sets, since the logical condition method adopted the idea of the representations of Boolean functions, norms were limited to a small range and could not be changed. Meanwhile, inequalities constructed by the convex-hull-based method usually had larger norms and could not be controlled by users. In [34], Feng *et al.* proposed a method to find the minimal norm based on theoretical research and realized the complete control of the norm, which led to the reduction of model-solving time.

1.2 Our Contributions

In this paper, we concentrate on the number of inequalities and the maximal absolute value of their coefficients in a full linear integer inequality characterization and construct a large number of experiments to explore the relationship between these two factors and the runtime of the MILP model.

First of all, we propose a MILP model to check whether a given set can be characterized by a linear inequality with coefficients in a given range. Based on this algorithm and the idea of binary search, a new algorithm to find an equivalent inequality of a given inequality with the minimal norm is proposed. Moreover, we also provide an algorithm to select a FLIIC with a given cardinality from the candidate set.

Based on the discussion of the norm and cardinality, we characterize basic components of block ciphers with equivalent inequality sets whose norms and cardinalities are different. Then we construct a large number of experiments and compare the runtimes of models to explore the effect of the norm and cardinality.

From the results of our experiments, it makes sense to say that reducing the scale of the model is helpful, although this assertion may not always be accurate. In the case that the relationship between the modeling method and the runtime cannot be thoroughly studied, the characterization with the minimal norm and cardinality is a better choice in general. The source codes and results of this paper are available at https://gitee.com/ShelwinXu/on-two-factors-affecting-the-efficiency-of-milp-models-in-automated-cryptanalyses-code.git.

1.3 Organization

The rest of the paper is organized as follows: some preliminaries and notations are given in Section 2. In Section 3, new algorithms to control the norm and cardinality of inequality sets are proposed as the theoretical basis of the experiment. Finally, in Section 4, a large number of experiments are conducted to study the effect of the norm and cardinality on the runtime of the model.

2 Preliminaries

In this section we will give a brief overview of notations and definitions used in this paper. Table 1 lists parts of notations.

Table 1. The notations used throughout the paper

Notation	Description
n	A positive integer
$\mathbb{Z}$	The set of all integers
$\mathbb{R}$	The set of all real numbers
$\mathbb{Z}_2$	The set $\{0, 1\}$
$\mathbb{Z}_2^n$	The set of all n-tuples over $\mathbb{Z}_2$, i.e., $\{0, 1\}^n$
$wt(x)$	Hamming weight of x
e_i	An n-bit unit whose i-th element is 1 and others are 0
$x \oplus y$	Bitwise XOR between x and y
S	A subset of $\mathbb{Z}_2^n$
$\overline{S}$	The complementary set of S in $\mathbb{Z}_2^n$
$l : \sum\limits_{i=0}^{n-1} a_i x_i \geq b$	A linear inequality whose coefficients are integers
$(a_0, a_1, \cdots, a_{n-1}, b)$	The linear inequality $\sum\limits_{i=0}^{n-1} a_i x_i \geq b$
$L = \{l_i \vert l_i : \sum\limits_{j=0}^{n-1} a_{i,j} x_j \geq b_i\}$	A set of inequalities whose coefficients are integers

2.1 MILP Models and Full Characterization

MILP is a fundamental method of finding the maximal or minimal value of a linear objective function whose variables are subjected to certain linear constraints (linear inequalities) and has been widely used in operations research, graph theory, computational geometry, *etc* [35]. A MILP problem usually consists of three parts: objective function, linear constraints and variables. It can be formally stated as follows: given $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$ and $c_1, \cdots, c_n \in \mathbb{R}$, find $x \in \mathbb{Z}^k \times \mathbb{R}^{n-k} \subseteq \mathbb{R}^n$ with $Ax \le b$ such that $c_1x_1 + c_2x_2 + \cdots + c_nx_n$ is maximized or minimized, where $\mathbb{R}$ and $\mathbb{Z}$ are the sets of all real numbers and all integers respectively. Linear constraints are a set of linear inequalities which can be writen as:

$$\begin{cases} a_{1,1}x_1 + \cdots + a_{1,n}x_n \ge b_1, \\ a_{2,1}x_1 + \cdots + a_{2,n}x_n \ge b_2, \\ \qquad\qquad \vdots \\ a_{m,1}x_1 + \cdots + a_{m,n}x_n \ge b_m. \end{cases}$$

In most MILP-based cryptanalyses, the variable x_i is usually set as a binary variable, that is, the value is limited in $\mathbb{Z}_2$, and elements in the coefficient matrix A corresponding to the constraint conditions are restricted to $\mathbb{Z}$. In the MILP model, the objective function can be set as maximization/minimization, or be omitted. When the model does not have an objective function, the output result is a feasible solution or no solution.

To construct MILP models for the cryptanalytic algorithm, attackers need to characterize basic components of ciphers by linear integer inequalities. By ensuring the solution space of these linear integer inequalities is the solution required for cryptanalyses, attackers can perform automated cryptanalyses by solving the MILP model. This kind of modeling method is called the full linear integer inequality characterization.

Definition 1 (FLIIC). *Let $S \subset \mathbb{Z}_2^n$ and L be a set of linear integer inequalities:*

$$\begin{cases} a_{0,0}x_0 + a_{0,1}x_1 + \cdots + a_{0,n-1}x_{n-1} + b_0 \ge 0, \\ a_{1,0}x_0 + a_{1,1}x_1 + \cdots + a_{1,n-1}x_{n-1} + b_1 \ge 0, \\ \qquad\qquad \vdots \\ a_{m-1,0}x_0 + a_{m-1,1}x_1 + \cdots + a_{m-1,n-1}x_{n-1} + b_{m-1} \ge 0, \end{cases} \tag{1}$$

where $a_{i,j}$ and b_i are integers for $0 \le i \le m-1, 0 \le j \le n-1$. L is called a full linear integer inequality characterization (FLIIC, in short) of S if the solution set of L on $\mathbb{Z}_2^n$ is S exactly. We also say L fully characterizes S, and m is called the cardinality of L, denoted by $|L|$.

We say L_1 and L_2 are equivalent if they fully characterize the same set S. In particular, l_2 is said to be an equivalent inequality of l_1 if its solution space is the same as l_1.

Define the norm $\parallel l \parallel$ of l as below:

$$\parallel l \parallel := \max\{|a_i|, |b|, 0 \le i \le n-1\}.$$

Similarly, let $L = \{l_i | 0 \le i \le m-1\}$, and we have $\mathbf{Sov}(L) = \bigcap_{i=0}^{m-1} \mathbf{Sov}(l)$, where $\mathbf{Sov}(l)$ means the solution set of the inequality l on $\mathbb{Z}_2^n$. Particularly, if L has only one inequality l, we can use l instead of L. The norm $\parallel L \parallel$ of L is defined as below:

$$\parallel L \parallel := \max\{\parallel l_i \parallel \;\big|\; 0 \le i \le m-1\}.$$

In this paper, we will focus on the effect of the norm and cardinality of a FLIIC on the efficiency of solving it in Gurobi.

2.2 Set Covering Problem

Set Covering Problem has important applications in the MILP-based automated search, which is used to choose as few inequalities as possible from the large candidate inequality set to assemble a FLIIC for a given set in $\mathbb{Z}_2^n$. First of all, a formal description of SCP is provided.

Suppose $\mathcal{U}$ is a given set, denote $\mathcal{S}$ as a subset of the power set of $\mathcal{U}$ which contains n elements whose union is $\mathcal{U}$. SCP is aimed to find a minimal subset of $\mathcal{S}$ such that their union is equal to $\mathcal{U}$. The specific description is as follows:
Decision variables:

$$y_s = \begin{cases} 1, & \text{if } s \in \mathcal{S} \text{ is selected;} \\ 0, & \text{otherwise.} \end{cases}$$

Objective function:

$$\text{Minimize} \sum_{s \in \mathcal{S}} y_s.$$

Constraint conditions:

$$\sum_{s:e \in s} y_s \geq 1, \forall e \in \mathcal{U};$$

$$y_s \in \{0, 1\}, \forall s \in \mathcal{S}.$$

The objective function is set as the minimal value of the number of selected subsets and the constraint conditions ensure that all elements in $\mathcal{U}$ are covered.

SCP has been proven to be NPC [36], which plays a very important role in reducing the scale of models for better characterization in automatic cryptanalyses. Traditional MILP characterization can be divided into two steps: the first step is to generate sufficient characterization inequalities for the given set and establish a full characterization as a candidate set; the second step is to remove redundant inequalities from this candidate set, i.e., select the minimal number of inequalities as the final full characterization. The second step can be completely converted into an SCP, where the set composed of all infeasible points is the target set $\mathcal{U}$, the set of points cut off by each candidate inequality is regarded as an element of $\mathcal{S}$, and all the candidate inequalities can be regarded as $\mathcal{S}$. The minimal number of inequality characterization can be obtained by selecting the minimal subset of $\mathcal{S}$ to cover $\mathcal{U}$. According to the definition, SCP is equivalent to a MILP which can be solved directly by MILP solvers and get the optimal value. This method is efficient for low dimensional sets and can provide a theoretical guarantee of optimality. However, due to the large number of candidate inequalities and infeasible points, the scale of a MILP model corresponding to SCP is usually large, and it is very difficult to obtain the optimal solution directly. In this case, other methods such as the greedy algorithm and heuristic algorithms are often adopted to get better solutions. Although it is difficult to prove the optimality theoretically, a good solution can still be obtained by solving an SCP for high dimensional sets.

3 Algorithms to Control the Norm and Cardinality

In this section, we propose a MILP model $\mathcal{M}(S, \overline{S}, \mathcal{B})$ to check if S can be characterized by a linear inequality l with $\parallel l \parallel \leq \mathcal{B}$ for a given positive integer $\mathcal{B}$, and a new algorithm $ReducedNorm(l)$ to find an equivalent inequality of l with the minimal norm. $ReducedNorm(l)$ can change coefficients of inequalities while keeping the solution space unchanged, which is very important to study the effect of the norm of FLIIC on the solving efficiency of the model. Meanwhile, an algorithm to select FLIICs of a given cardinality is also provided in this section.

3.1 Algorithm to Construct a FLIIC with a Specified Range of Coefficients

First, we propose a MILP model $\mathcal{M}$ without an objective function to determine whether a given set can be characterized by an inequality with coefficients in a specified range:

$\mathcal{M}(S, \overline{S}, \mathcal{B})$ **Variables.** There are $n+1$ integer variables in this MILP model, $a_0, a_1, \cdots, a_{n-1}, b$.

$\mathcal{M}(S, \overline{S}, \mathcal{B})$ **Bound.** These $n+1$ variables are all integers and bounded by $\mathcal{B}$:

$$-\mathcal{B} \leq a_i \leq \mathcal{B}, i = 0, 1, 2, \cdots, n-1,$$

$$-\mathcal{B} \leq b \leq \mathcal{B}.$$

$\mathcal{M}(S, \overline{S}, \mathcal{B})$ **Constraints.** Since $a_0, a_1, \cdots, a_{n-1}, b$ represent the coefficients of our target linear inequality characterization of S, we have the following constraints:

For $(x_0, x_1, \cdots, x_{n-1}) \in S$,

$$\sum_{i=0}^{n-1} a_i x_i \geq b.$$

For $(y_0, y_1, \cdots, y_{n-1}) \in \overline{S}$,

$$\sum_{i=0}^{n-1} a_i y_i \leq b - 1.$$

Algorithm 1 is proposed to construct $\mathcal{M}(S, \overline{S}, \mathcal{B})$ and solve it. As a result, if $\mathcal{M}(S, \overline{S}, \mathcal{B})$ is feasible, then we get a FLIIC l of S, where $l = (a_0, a_1, \cdots, a_{n-1}, b)$ with $\parallel l \parallel \leq \mathcal{B}$.

Moreover, we have a basic observation that can be used to construct a FLIIC whose norm is just closely to a given value:

Observation 1 *In Gurobi, when there is no objective function, the feasible solution output by the solver tends to be close to the boundary.*

Algorithm 1 $\mathcal{M}(S, \overline{S}, \mathcal{B})$

```
Input: $S, \overline{S} \in \mathbf{\Pi}_n$, a positive integer $\mathcal{B}$
Output: $n+1$ integers $(a_0, a_1, \cdots, a_{n-1}, b)$ or INFEASIBLE
Declare an empty MILP mode $\mathcal{M}$;
$\mathcal{M}.Var \leftarrow a_i \in [-\mathcal{B}, \mathcal{B}], i \in \{0, 1, \cdots, n-1\}$;
$\mathcal{M}.Var \leftarrow b \in [-\mathcal{B}, \mathcal{B}]$;
for all points $(x_0, x_1, \cdots, x_{n-1}) \in S$ do
    $\mathcal{M}.Con \leftarrow a_0x_0 + a_1x_1 + \cdots + a_{n-1}x_{n-1} - b \geq 0$ ;
end for
for all points $(y_0, y_1, \cdots, y_{n-1}) \in \overline{S}$ do
    $\mathcal{M}.Con \leftarrow a_0y_0 + a_1y_1 + \cdots + a_{n-1}y_{n-1} - b \leq -1$ ;
end for
$\mathcal{M}.Optimize()$;
if $\mathcal{M}$ is infeasible then
    return INFEASIBLE;
else
    return $(a_0, a_1, \cdots, a_{n-1}, b)$;
end if
```

For example, while S and $\overline{S}$ are the same, the solution of $\mathcal{M}(S, \overline{S}, 50)$ is

$$l_0 = (50, -38, 50, 25, -12, 26, 24, 11, 0),$$

whose norm is 50, and the solution of $\mathcal{M}(S, \overline{S}, 500)$ is

$$l_1 = (-300, 99, 201, -100, 201, -300, 500, -100, 500),$$

whose norm is 500. According to this observation, we can control norms by setting different bounds for variables in $\mathcal{M}$. We take the block cipher Present as an example, and illustrate the relationship between norms of FLIICs and mean absolute values of coefficients generated by this method in Table 2. It can be seen that a uniform amplification of the coefficients is achieved and FLIICs with different norms can be generated.

Table 2. The norm and mean value of the coefficients of the FLIIC of Present's S-box

Norm	19	50	100	500	1000	5000	10000	50000
Mean Value	4.31	23.62	47.47	238.51	477.34	2387.94	4776.14	23882.02

3.2 Algorithm to Obtain a FLIIC with the Minimal Norm

It is worth mentioning that although it is difficult to quantify the relationship between the solving speed of a MILP solver and the norm of inequalities in automated search algorithms, people tend to select a smaller norm to make the solving speed faster. Therefore, we propose a norm reduction algorithm with the help of the MILP solver and the binary search algorithm. With the new algorithm, we can find an equivalent one with the minimal norm for a given inequality. It also provides a new way to construct a MILP model with a smaller scale. Next, we will show how to reduce the norm of linear inequalities in detail.

For a given linear inequality $l : \sum_{i=0}^{n-1} a_i x_i \geq b$, the solution set of l is computed and denoted as S, the norm of l is denoted as $\mathcal{B}$. We know that the model $\mathcal{M}(S, \overline{S}, \mathcal{B})$ is feasible since $(a_0, a_1, \cdots, a_{n-1}, b)$ is a solution of this model. $ReducedNorm(l)$ is used to find the minimal $\mathcal{B}_{min}$ such that $\mathcal{M}(S, \overline{S}, \mathcal{B}_{min})$ is also feasible. With the idea of the binary search, we turn to check whether the model $\mathcal{M}(S, \overline{S}, \mathcal{B}/2)$ is feasible. If a feasible solution is returned, then we can find an equivalent inequality l' with $\parallel l' \parallel \leq \mathcal{B}/2$. Otherwise, we need to compute the model $\mathcal{M}(S, \overline{S}, 3\mathcal{B}/4)$. Repeat the above process, then an equivalent inequality of l with the minimal norm can be obtained. The new algorithm needs to call the MILP solver $\mathcal{O}(\log(\mathcal{B}))$ times. For more details, please refer to Algorithm 2.

Algorithm 2 $ReducedNorm(l)$: Algorithm of reducing the norm of a given linear inequality l

Input: An initial inequality $l : a'_0 x_0 + a'_1 x_1 + \cdots + a'_{n-1} x_{n-1} \geq b'$.
Output: An equivalent inequality of l with the minimal norm.

```
for all points $(x_0, x_1, \cdots, x_{n-1}) \in \mathbb{Z}_2^n$ do
    if $a'_0 x_0 + a'_1 x_1 + \cdots + a'_{n-1} x_{n-1} \geq b'$ then
        $S \leftarrow (x_0, x_1, \cdots, x_{n-1})$;
    else
        $\overline{S} \leftarrow (x_0, x_1, \cdots, x_{n-1})$;
    end if
end for
$low := 0$;
$high := max\{abs(a'_i), abs(b'), i = 0, 1, \cdots, n-1\}$;
while $low \leq high$ do
    $mid = (low + high)/2$;
    $\mathcal{M}.Optimize()$;
    if $\mathcal{M}(S, \overline{S}, mid)$ is feasible then
        $l' \leftarrow$ the solution of $\mathcal{M}(S, \overline{S}, mid)$;
        $high = mid - 1$;
    else
        $low = mid + 1$;
    end if
end while
return $l'$;
```

3.3 Algorithm to Select a FLIIC with a Given Cardinality

Recall the previous introduction in subsection 2.2, we know that an SCP can be completely converted into a MILP problem, hence it can be solved by MILP solvers. In this subsection, we introduce a MILP-based algorithm to build a FLIIC from a given candidate set and provide a parameter to determine its cardinality, please refer to Algorithm 3 for more details.

Algorithm 3 *SelectIneq*$(\mathcal{L}, N)$: Algorithm of selecting the FLIIC of a given cardinality N

```
Input: A given candidate set $\mathcal{L} = \{L_i | i = 0, \cdots, n-1\}$ and a given integer $N$.
Output: A FLIIC whose cardinality is $N$.
$\mathcal{U} = \{u_0, \cdots, u_{k-1}\} \leftarrow$ the infeasible points of $\mathcal{L}$;
for $i = 0$ to $n-1$ do
    $U_i \leftarrow$ the infeasible points of $L_i$;
end for
Declare an empty MILP mode $\mathcal{M}$;
$\mathcal{M}.Var \leftarrow z_i \in \{0,1\}$, $i = 0, \cdots, n-1$;
for $i = 0$ to $k-1$ do
    $\mathcal{M}.Con \leftarrow a_0 z_0 + a_1 z_1 + \cdots + a_{n-1} z_{n-1} \geq 1$, if $u_i \in U_j$ then $a_j = 1$, else $a_j = 0$;
end for
$\mathcal{M}.Optimize()$;
if $\mathcal{M}$ is infeasible then
    return INFEASIBLE;
else
    return $\mathcal{L}' = \{L_{z_i} | z_i = 1$ in the solution of $\mathcal{M}\}$;
end if
```

Algorithm 3 constructs a FLIIC with a given cardinality N by solving a MILP model without an objective function. Moreover, by adding an objective function to minimize the number of selected inequalities, a FLIIC with the minimal cardinality can be obtained. The same goal can be achieved by iterating N incrementally from 0 and solving *SelectIneq*$(\mathcal{L}, N)$ until it is feasible for the first time.

4 Experimentally Based Observations of the Effects of the Norm and Cardinality

In this section, based on the idea of the control variable method, we construct a large number of samples for automatic search models, and explore the effect of the norm and cardinality through the experimental results.

In specific experiments, we keep the overall framework of the automated search model unchanged and only replace the FLIIC of basic components such as S-boxes to achieve the purpose of controlling variables. The relationship between the two target parameters and the solving efficiency of the model is investigated by using FLIICs with different norms and cardinalities.

4.1 The Setting of Controling Variables

To explore the influence of the norm and cardinality on the runtime of MILP models, models with different norms/cardinalities are generated by constructing FLIICs of S-boxes with different norms/cardinalities in SPN block ciphers. Meanwhile, other details are kept the same in these models.

For the norm, we first call Algorithm 2 to obtain a new FLIIC with the minimal norm from a given FLIIC and denote the model using this FLIIC as Test 0. Next, based on Observation 1, equivalent inequalities of the inequalities used in Test 0 with different norms are constructed by calling Algorithm 1.

For the cardinality, we control it by setting different N in $SelectIneq(\mathcal{L}, N)$ which is introduced in Subsection 3.3. The candidate set $\mathcal{L}$ stores inequalities to characterize a given S-box which can be generated by previous methods and is fixed for all N. First, the FLIIC with the minimal cardinality in the sense of $\mathcal{L}$ is obtained for the Test 0 model. Then FLIICs with different cardinalities are constructed to characterize the given S-box.

4.2 The Method of Generating Samples

Our motivation is to conduct a large number of experiments to reflect statistical regularities. Since the background is the automatic search for differential trails, it is hard to carry out thousands of experiments based on practical ciphers. To deal with this issue, many new examples without deviating from the actual application scenario are constructed. In our experiments, two kinds of automatic search models are considered: the search of the minimal number of differential active S-boxes and the search of impossible differential trails, for the cases with and without objective functions and denoted them by Type 0 and Type 1 respectively.

For Type 0 tests, based on models of SPN structures, different S-layers and P-layers are combined to produce new samples. For the S-layer, we select different 4-bit S-boxes used in block ciphers such as Present, Rectangle and Lillput. For the P-layer, we make small changes to the bit-permutation layers of existing ciphers, for example, by swapping two positions in the 64-bit bit-permutation layer of Present, $\binom{64}{2} = 2016$ new samples can be generated.

For Type 1 tests, exhaustive search is adopted as the traditional search of impossible differential trails. More specifically, the input and output differences are fixed to determine whether the differential trail is feasible or not. For 64-bit block ciphers, when hamming weights of the input difference x and the output difference y are restricted to 1, i.e., $wt(x) = wt(y) = 1$, all possible combinations of them need to be exhausted, which means that $64 \times 64 = 4096$ new samples are constructed.

Sufficient samples can be generated based on the above two types of experiments. Since these samples are abstracted from specific automated cryptanalyses, they can reflect the efficiency of solving models for automated cryptanalyses convictively. Besides, different rounds of automatic search also correspond to different samples. Combined with the different modes of the norm and cardinality, the experimental modes are denoted as (T_i, R_i, N) and (T_i, R_i, C) respectively, where T_i means the Type i test, R_i means the i rounds search, N and C represent the norm and cardinality respectively.

4.3 Experimental Results and Discussion

For each generated sample, we conduct 8 experiments for the two parameters mentioned above. In more detail, for norms, we investigate the minimal norm, 50, 100, 500, 1000, 5000, 10000, and 50000, and denoted as Test 0, Test 1, $\cdots$, Test 7, respectively. For example, Present(T_0, R_3, N) Test 0 represents the model with the minimal norm which explores the effect of the norm for 3 round impossible differential search of samples with Present structure. For cardinalities, we investigated the minimal cardinality $N_{min}, N_{min} + 10, \cdots, N_{min} + 70$, and denoted as Test 0, Test 1, $\cdots$, Test 7 respectively. During the experiments, we record runtimes of each model for further statistical analysis.

In the experiments for each sample, we design the following statistics based on the recorded runtime when different variables are adopted:

- The number of Test 0 dominant samples (#Test 0 Dom): Among the experiments corresponding to each sample, if the runtime of Test 0 is the minimal, the sample is called Test 0 dominant. Similarly, %Test 0 Dom represents the percentage of samples that are Test 0 dominant;
- The mean of the best tests (BT Mean): Among the experiments of each sample, denote i as the best test if the runtime of Test i is the minimal, then the mean of the best tests reflects the approximate sequence number of the test which may lead to the minimal runtime;
- The optimal choice (Opt Choice): If the number of samples with the minimal runtime is the maximum under Test i for the current set of samples, then i is denoted as the optimal choice.

Since we pay more attention to the effect of the minimal norm and cardinality on the improvement of solution efficiency, #Test 0 Dom is designed to investigate the advantages of the minimal norm/cardinality. Meanwhile, if BT Mean is close to 0, it means that the smaller the norm/cardinality, the better the acceleration effect. The Opt Choice provides the best choice of the norm and cardinality in the overall consideration. Even if sometimes the percentage of samples that reach the minimal runtime under this choice is not exceptionally high, it still guarantees that there is no better choice.

See the detailed results of experiments in Table 3. Moreover, we also construct several automated search models for some lightweight block ciphers and draw bar charts for their runtimes, see Figure 1 and Figure 2.

Due to the small sample size, rules are unclear in Figure 1 and Figure 2. Even in this case, it can be seen that the FLIIC with the minimal norm and the minimal cardinality is a good choice, although there are some counter-examples. Indeed, sometimes it is hard to determine which choice is the best in advance for a model. However, according to a large amount of experimental data in Table 3,

Table 3. The experimental result of automated search models with different norms/cardinalities

Test	#Samples	#Test 0 Dom	%Test 0 Dom	BT Mean	Opt Choice
Present(T_0, R_3, N)	6051	3280	54.2%	1.97	0
Present(T_0, R_3, C)	6051	4655	76.93%	0.32	0
Rectangle(T_0, R_3, N)	6051	3538	58.4%	1.77	0
Rectangle(T_0, R_3, C)	6051	4995	82.5%	0.18	0
Present(T_1, R_{20}, N)	4096	3517	85.86%	0.51	0
Present(T_1, R_{25}, N)	4096	3359	82.01%	0.74	0
Present(T_1, R_{30}, N)	4096	3438	83.94%	0.58	0
Present(T_1, R_{20}, C)	4096	3978	97.12%	0.04	0
Present(T_1, R_{25}, C)	4096	3837	93.68%	0.08	0
Present(T_1, R_{30}, C)	4096	3747	91.48%	0.12	0
Rectangle(T_1, R_{20}, N)	4096	4076	99.51%	0.02	0
Rectangle(T_1, R_{25}, N)	4096	4084	99.71%	0.02	0
Rectangle(T_1, R_{30}, N)	4096	4075	99.49%	0.02	0
Rectangle(T_1, R_{20}, C)	4096	4096	100.00%	0	0
Rectangle(T_1, R_{25}, C)	4096	4096	100.00%	0	0
Rectangle(T_1, R_{30}, C)	4096	4096	100.00%	0	0
Total	73356	62867	85.70%	-	0

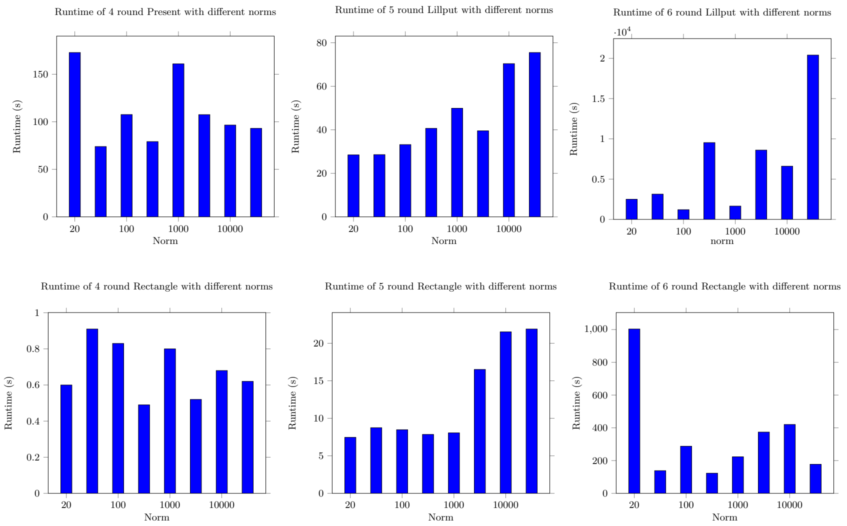

Fig. 1. Influence of the norm on the runtime

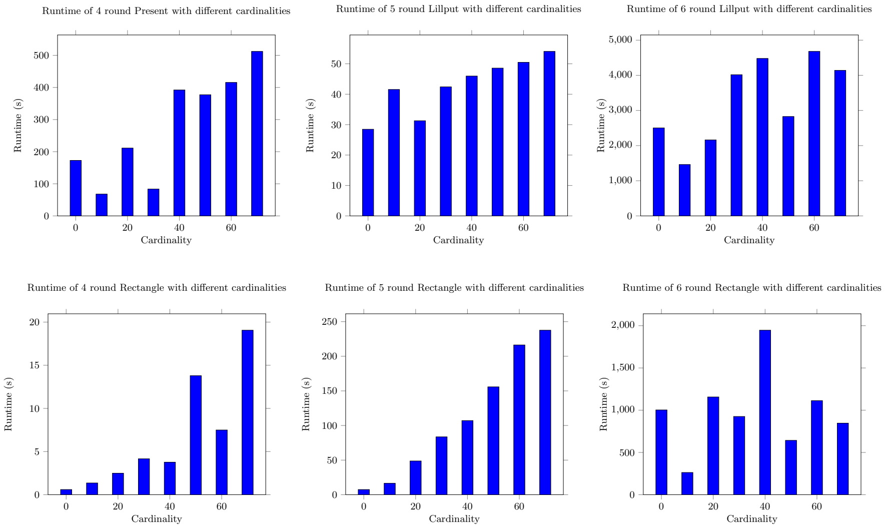

Fig. 2. Influence of the cardinality on the runtime

it is a better choice to use the FLIIC with the minimal norm and the minimal cardinality in most cases.

Acknowledge

This research was supported by National Key Research and Development Project under Grant No. 2018YFA0704705 and CAS Project for Young Scientists in Basic Research (Grant No. YSBR-035).

References

1. Sherif A Fahmy. Milp model and a genetic algorithm for the integrated cell formation, layout and worker assignment problem. In *2019 IEEE 10th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT)*, pages 117–121. IEEE, 2019.
2. Bilge Bilgen and Hans-Otto Günther. A milp model for production and distribution planning in consumer goods supply chains. In *Operations research proceedings 2008*, pages 179–184. Springer, 2009.
3. Xin Wang, Jason Atkin, Serhiy Bozhko, and Christopher Hill. Application of a milp-based algorithm for power flow optimisation within more-electric aircraft electrical power systems. In *2019 21st European Conference on Power Electronics and Applications (EPE'19 ECCE Europe)*, pages P–1. IEEE, 2019.
4. Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and linear cryptanalysis using mixed-integer linear programming. In *International Conference on Information Security and Cryptology*, pages 57–76. Springer, 2011.

5. Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. *Journal of CRYPTOLOGY*, 4(1):3–72, 1991.
6. Mitsuru Matsui. Linear cryptanalysis method for des cipher. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 386–397. Springer, 1993.
7. Lars R Knudsen. Truncated and higher order differentials. In *International Workshop on Fast Software Encryption*, pages 196–211. Springer, 1994.
8. Eli Biham. New types of cryptanalytic attacks using related keys. *Journal of Cryptology*, 7(4):229–246, 1994.
9. Eli Biham, Alex Biryukov, and Adi Shamir. Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 12–23. Springer, 1999.
10. Andrey Bogdanov and Vincent Rijmen. Zero-correlation linear cryptanalysis of block ciphers. *IACR Cryptology ePrint Archive*, 2011:123, 01 2011.
11. Siwei Sun, Lei Hu, Ling Song, Yonghong Xie, and Peng Wang. Automatic security evaluation of block ciphers with s-bp structures against related-key differential attacks. In *Lin D., Xu S., Yung M. (eds) Information Security and Cryptology. Inscrypt 2013. Lecture Notes in Computer Science, vol 8567*, pages 39–51. Springer, Cham, 2014.
12. Siwei Sun, Hu Lei, Wang Peng, Kexin Qiao, and Song Ling. Automatic security evaluation and (related-key) differential characteristic search: Application to simon, present, lblock, des(l) and other bit-oriented block ciphers. In *Sarkar P., Iwata T. (eds) Advances in Cryptology - ASIACRYPT 2014. ASIACRYPT 2014. Lecture Notes in Computer Science, vol 8873*, pages 158–178. Springer, Berlin, Heidelberg, 2014.
13. Kai Fu, Meiqin Wang, Yinghua Guo, Siwei Sun, and Lei Hu. Milp-based automatic search algorithms for differential and linear trails for speck. In *International Conference on Fast Software Encryption*, pages 268–288. Springer, 2016.
14. Danping Shi, Siwei Sun, Yu Sasaki, Chaoyun Li, and Lei Hu. Correlation of quadratic boolean functions: Cryptanalysis of all versions of full morus. In *Boldyreva A., Micciancio D. (eds) Advances in Cryptology - CRYPTO 2019. CRYPTO 2019. Lecture Notes in Computer Science, vol 11693*, pages 180–209. Springer, Cham, 2019.
15. Cui Tingting, Chen Shiyao, Fu Kai, Wang Meiqin, and Jia Keting. New automatic search tool for impossible differentials and zero-correlation linear approximations. *SCIENCE CHINA Information Sciences*, 2016.
16. Yu Sasaki and Yosuke Todo. New impossible differential search tool from design and cryptanalysis aspects. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 185–215. Springer, 2017.
17. Julia Borghoff, Lars R.Knudsen, and Mathias Stolpe. Bivium as a mixed-integer linear programming problem. In *Parker M.G. (eds) Cryptography and Coding. IMACC 2009. Lecture Notes in Computer Science, vol 5921*. Springer, Berlin, Heidelberg, 2009.
18. Martin Albrecht and Carlos Cid. Cold boot key recovery by solving polynomial systems with noise. In *Lopez J., Tsudik G. (eds) Applied Cryptography and Network Security. ACNS 2011. Lecture Notes in Computer Science, vol 6715*, pages 57–52. Springer, Berlin, Heidelberg, 2011.
19. Michael Walter, Stanislav Bulygin, and Johannes Buchmann. Optimizing guessing strategies for algebraic cryptanalysis with applications to epcbc. In *Kutyłowski M., Yung M. (eds) Information Security and Cryptology. Inscrypt 2012. Lecture Notes in Computer Science, vol 7763*, pages 175–197. Springer, Berlin, Heidelberg, 2013.

20. Zejun Xiang, Wentao Zhang, Zhenzhen Bao, and Dongdai Lin. Applying milp method to searching integral distinguishers based on division property for 6 lightweight block ciphers. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 648–678. Springer, 2016.
21. Yosuke Todo, Takanori Isobe, Yonglin Hao, and Willi Meier. Cube attacks on non-blackbox polynomials based on division property. *IEEE Transactions on Computers*, 67(12):1720–1736, 2018.
22. Zhe Cen, Xiutao Feng, Zhangyi Wang, and Chunping Cao. Minimizing deduction system and its application. *arXiv preprint arXiv:2006.05833*, 2020.
23. LLC Gurobi Optimization. Gurobi optimizer reference manual. 2020.
24. IBM ILOG Cplex. V12. 1: User's manual for cplex. *International Business Machines Corporation*, 46(53):157, 2009.
25. Niklas Eén and Niklas Sörensson. An extensible sat-solver. In *International conference on theory and applications of satisfiability testing*, pages 502–518. Springer, 2003.
26. Yu Sasaki and Yosuke Todo. New algorithm for modeling s-box in milp based differential and division trail search. In *Farshim P., Simion E. (eds) Innovative Security Solutions for Information Technology and Communications. SecITC 2017. Lecture Notes in Computer Science, vol 10543*, pages 150–165. Springer, Cham, 2017.
27. Lingchen Li, Wenling Wu, Yafei Zheng, and Lei Zhang. The relationship between the construction and solution of the milp models and applications. *Cryptology ePrint Archive*, 2019.
28. Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song. Automatic security evaluation and (related-key) differential characteristic search: application to simon, present, lblock, des (l) and other bit-oriented block ciphers. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 158–178. Springer, 2014.
29. Sage Developers. Sagemath, the sage mathematics software system (version 9.0). 2020.
30. Ahmed Abdelkhalek, Yu Sasaki, Yosuke Todo, Mohamed Tolba, and Amr M Youssef. Milp modeling for (large) s-boxes to optimize probability of differential characteristics. *IACR Transactions on Symmetric Cryptology*, pages 99–129, 2017.
31. Christina Boura and Daniel Coggia. Efficient milp modelings for sboxes and linear layers of spn ciphers. *IACR Transactions on Symmetric Cryptology*, pages 327–361, 2020.
32. Aleksei Udovenko. Milp modeling of boolean functions by minimum number of inequalities. Cryptology ePrint Archive, Paper 2021/1099, 2021. `https://eprint.iacr.org/2021/1099`.
33. Ting Li and Yao Sun. Superball: A new approach for milp modelings of boolean functions. *IACR Transactions on Symmetric Cryptology*, 2022, Issue 3:341–367, 2022.
34. Xiutao Feng, Yu Tian, Yongxing Wang, Shengyuan Xu, and Anpeng Zhang. Full linear integer inequality characterization of sets over $\mathbb{Z}_2^n$. `http://www.chinaxiv.org/abs/202210.00055v2`.
35. Kantorovich and Leonid Vitalievich. A new method of solving some classes of extremal problems. In *Dokl. Akad. Nauk SSSR 28, 3*, pages 211–214, 1940.
36. Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.